



Service – Oriented Architecture

Web-Service-Technologien

Sebastian Reese



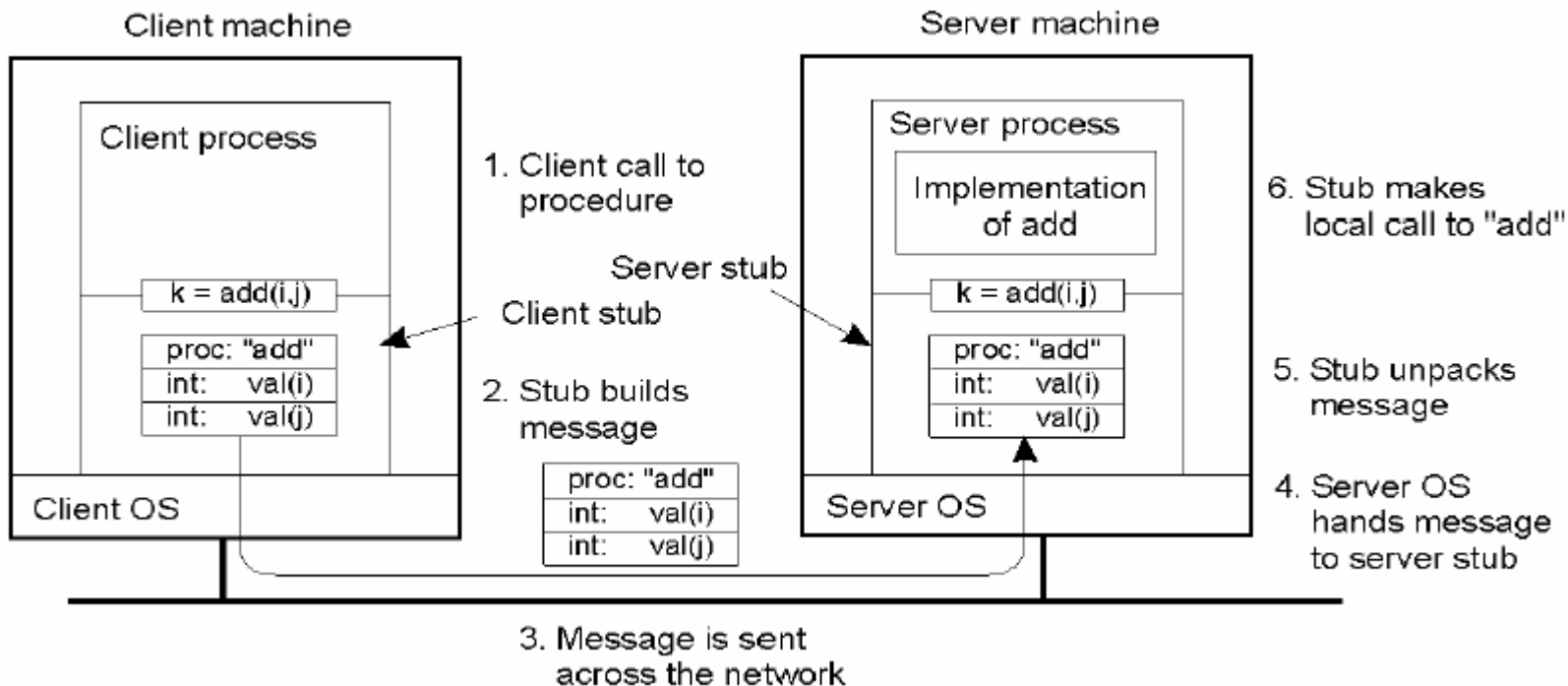
- Einleitung
 - Entfernte Aufrufe
 - Was ist ein Service
 - XML Web-Service
 - Rollen von Web-Services
 - Serviceorientierte Softwarearchitektur
 - Forderungen an Serviceorientierung
 - Integrationlayer
 - Rollen von Web-Services II
- Web-Services (1. Generation)
 - WSDL
 - SOAP
 - UDDI



- Web-Services (2. Generation)
 - Neuerungen der 2. Generation
 - Koordination und Transaktion
 - BPEL4WS
 - Sicherheit
 - Reliable Messaging
 - Policy
 - Attachments

→ Entfernte Aufrufe

- Remote Procedure Call (RPC)
- Methodenaufruf über ein Transportnetz





→ Was ist ein Service

- Eigenständige unabhängige Softwarekomponente
- Geschäftsfunktionalität
- Lose Kopplung



- Forderungen an XML Web-Services
 - Nachrichtenaustausch:
 - via Internet (z.B. HTTP)
 - Nachrichtenformat:
 - XML
 - (Plattform- und Sprachunabhängig)



- allgemeine Norm für Web-Services (de facto Standard)
- Service Beschreibungsdatei (WSDL Dokument)
- Nachrichten als SOAP Envelope via HTTP
- Allgemeiner Suchdienst
- Agieren als „Requestor“ und „Provider“

→ Rollen von Web-Services

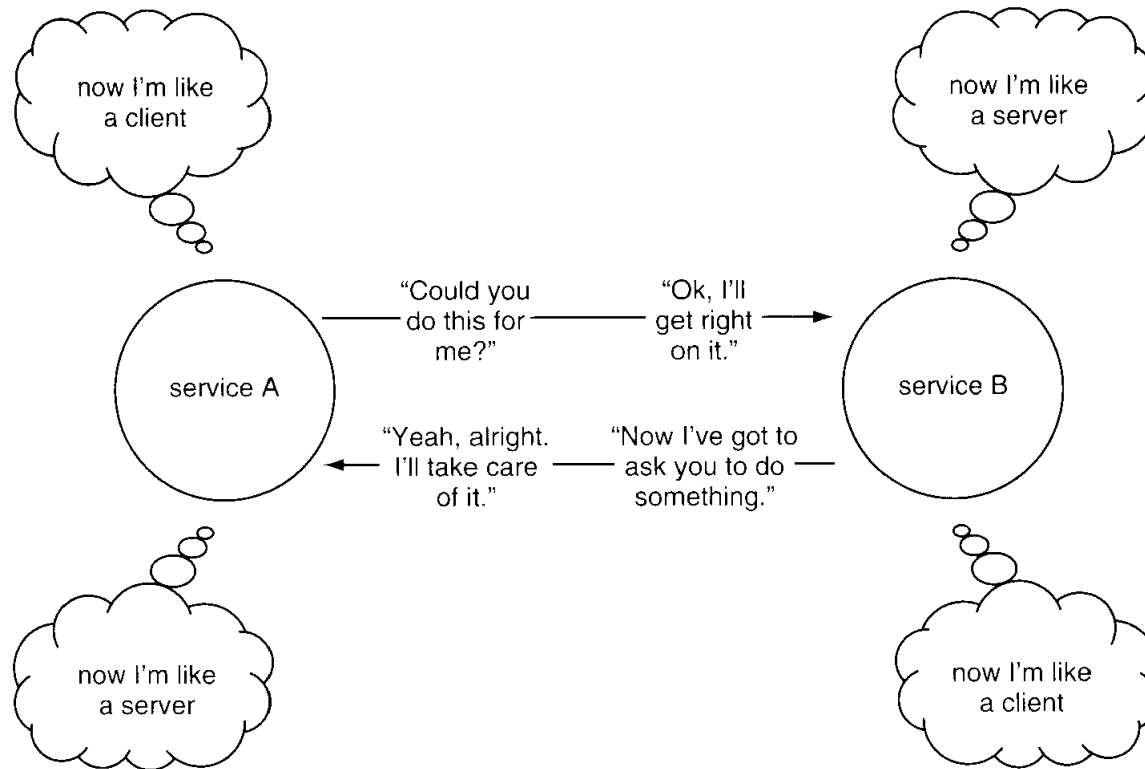


Figure 3.2
Web services swapping roles during a conversation.



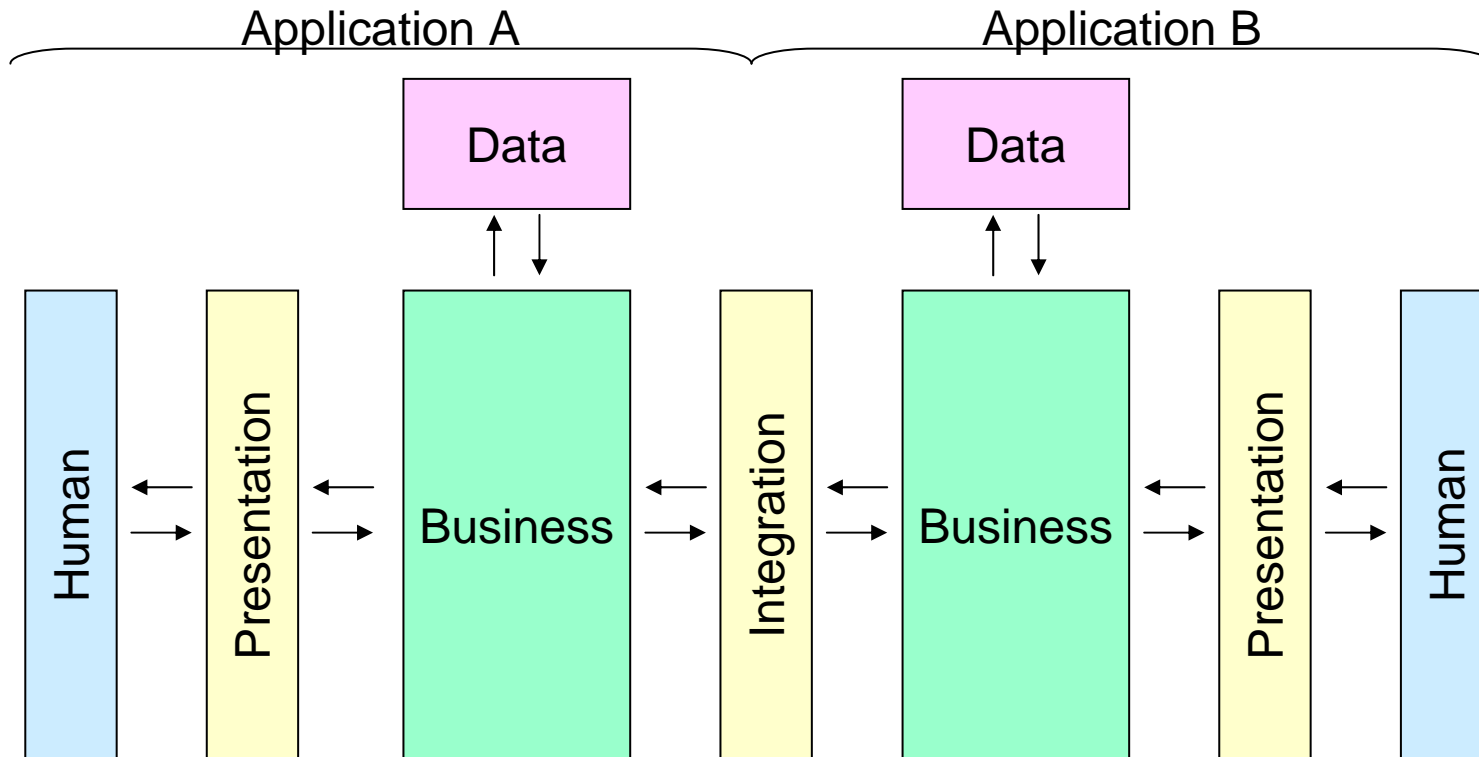
- ~~Anwendungen die Web-Services nutzen~~
- Anwendungen die auf Serviceorientiertheit basieren
 - Kapselung von Businesslogik in Services
 - Bindung an Web-Service Prinzipien
 - Einführen einer neuen Schicht



→Forderungen an Serviceorientierung

- Wiederverwendbarkeit / Aufteilung
- Formale Bedingungen
- Lose Kopplung zwischen Services
- Nur Service Beschreibung bekannt
- Services sind kombinierbar
- Autonomie
- Statuslosigkeit
- Auffindbar von Entwicklern über Suchdienst

→ Integrationlayer



- Integrationlayer als Presentationlayer für Maschinen



→ Rollen in Web-Services II

- Service provider
- Service requestor
- Intermediary
- Initial Sender
- Ultimate receiver



1st Generation Web-Services

WSDL

SOAP

UDDI



- Web-Service Description Language
 - Beschreibung von Web-Services
 - Was macht der WS?
 - Wo kann der WS erreicht werden?
 - Wie muss WS angesprochen werden?
- Beispiel Googlesearch.xml



→WSDL II

```
<definitions>
  <!-- abstract part -->
  <types>
    <!-- type definition -->
  </types>
  <messages>
    <!-- declaration of message (parameters) -->
  </messages>
  <interface>
    <!-- declare operations and associate with messages (parameters) -->
  </interface>

  <!-- Concrete part -->
  <binding>
    <!-- associate operations with message format and protocol -->
  </binding>
  <service>
    <!-- definition of endpoints with physically addresses of SOAP nodes -->
  </service>
</definitions>
```

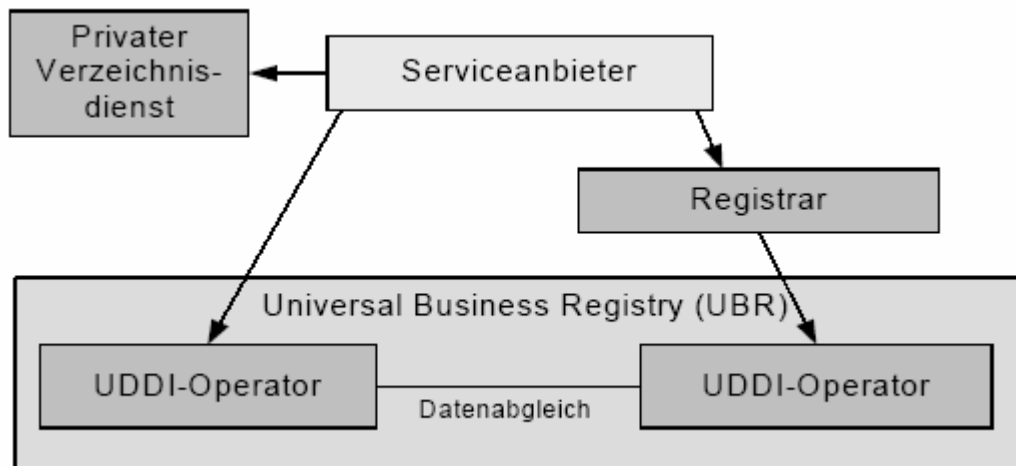


- Simple Object Access Protocol
- Nicht
Service-Oriented Architecture Protocol
- Standard Nachrichtenformat für
entfernte Aufrufe RPC



- SOAP Node (auch listener, server)
- Rollen
 - Initial Sender
 - Ultimate receiver
 - intermediary
- SOAP Envelope
 - Optional <header />
 - Required <body />

- Universal Description, Discovery and Integration





- **Verschiedene Arten von Registries**
 - **Public business registry**
 - UDDI- oder auch Node Operator
 - **Service Marketplaces**
 - Bezahl Node-Operator
 - **Private Registries**
 - Organisationsintern und für ausgewählte Businesspartner
 - **Internal Registry**
 - nur Organisationsintern



```
<businessEntity businessKey="e9355d51-32ca-49cf-8eb4-1ce59afbf4a7"  
  operator="Microsoft Corporation"  
  authorizedName="Thomas Erl">  
  <discoveryURLs>  
    <discoveryURL useType="businessEntity">  
      http://test.uddi.microsoft.com/discover?businesskey=e9355d51-32ca-  
49cf-8eb4-1ce59afbf4a7  
    </discoveryURL>  
  </discoveryURLs>  
  <name xml:lang="en">  
    XMLTC Consulting Inc  
  </name>  
  <description xml:lang="en">  
    XMLTC has been building end-to-end enterprise  
    eBusiness solutions for corporations and  
    government agencies since 1996. We offer a  
    wide range of design, development and  
    integration services  
  </description>
```



```
<businessServices>
  <businessService
    serviceKey="leeecfal-6f99-460e-a392-8328d38b763a"
    businessKey="e9355d51-32ca-49cf-8eb4-1ce59afbf4a7">
    <name xml:lang="en-us">
      Corporate Home Page
    </name>
    <bindingTemplates>
      <bindingTemplate
        bindingKey="48b02d40-0312-4293-a7f5-4449ca190984"
        serviceKey="leeecfal-6f99-460e-a392-8328d38b763a">
        <description xml:lang="en">
          Entry point into the XMLTC Web site through which a number of resource sites can be accessed
        </description>
        <accessPoint URLType="http">
          http://www.xmltc.com/
        </accessPoint>
        <tModelInstanceDetails />
      </bindingTemplate>
    </bindingTemplates>
    <categoryBag />
  </businessService>
</businessServices>
</businessEntity>
```



Second-generation Web- Services

(WS-*) Web-Services



→ Neuerungen der 2. Generation

- Kontext und Transaktionen
- Geschäftsprozesse
- Sicherheit
- gesicherte Nachrichtenversendung
- Richtlinien
- Anhänge



- Einführung eines Koordinatorprozesses
- Ermöglicht
 - Atomare Transaktionen
 - ACID Prinzip
 - Geschäftsaktivitäten
 - „long-time“ Transaktionen



- Atomare Transaktionen
 - Bereitstellung von ACID funktionalität
 - „Commit“
 - „Rollback“
 - Sperren von Ressourcen



- Geschäftsaktivitäten
 - Langzeit Transaktionen
 - Problem: Ressourcen können nicht über langen Zeitraum gesperrt bleiben
 - Lösung: Kontext bleibt erhalten, aber Ressourcen werden freigegeben



- Fehlerbehandlung
 - Transaktion
 - einfaches Rollback
 - Aktivität
 - Start eines Kompensationsprozesses um Aktivität rückgängig zu machen



- Erstellen eines Kontextes

```
<CreateCoordinationContext>
  <ActivationService>
    <wsu:Address>
      http://www.examples.ws/activation
    </wsu:Address>
  </ActivationService>
  <RequestorReference>
    <wsu:Address>
      http://www.xmltc.com
    </wsu:Address>
  </RequestorReference>
  <CoordinationType>
    http://schemas.xmlsoap.org/ws/2002/08/wstx
  </CoordinationType>
</CreateCoordinationContext>
```



- Skriptsprache
 - Strukturierte und Koordinierte Aufrufe von Web-Services
 - Beschrieben als WSDL-Datei



- Atomare Elemente

- Receive
- Invoke
- Reply
- Throw
- Wait

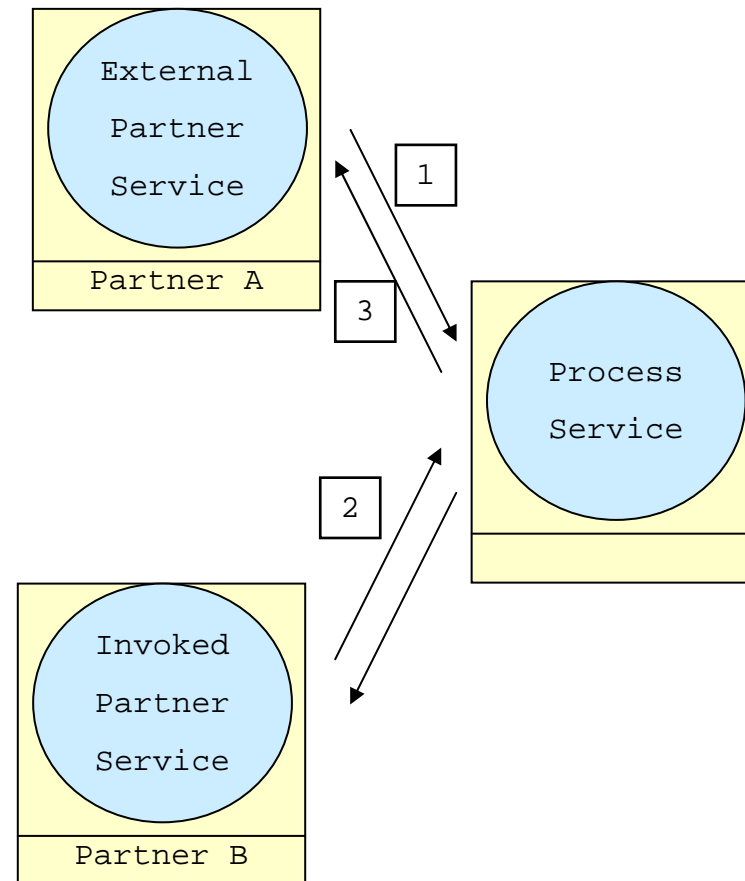
- Komplexe Elemente

- Sequence
- Flow
- Switch
- While

→BPEL4WS III

```

<process name="MyProcess">
  <partnerLinks/>
  <variables/>
  <faultHandlers/>
  <sequence>
    <recieve partner="PartnerA">
      ...
    </recieve>
    <invoke partner="PartnerB">
      ...
    </invoke>
    <reply partner="PartnerA">
      ...
    </reply>
  </sequence>
</process>
  
```





→ Sicherheit I

- Sicherheit in WS und XML relativ neu
- Benutzung von etablierten Techniken zur Erreichung von Sicherheit



- Sicherheitsproblem der
 - Identifikation
 - Authentifizierung
 - Autorisation
 - Integrität
 - Diskretion



→ Sicherheit III

Identifikation	WS-Security Framework
Authentifizierung	Extensible Access Control Markup Language (XACML)
Autorisation	Extensible Rights Markup Language (XrML) XML Key Management (XKMS) Security Assertion Markup Language (SAML) .NET Passport
Integrität	WS-Security Framework XML-Encryption Secure Socket Layer (SSL)
Diskretion	WS-Security Framework



→Reliable Messaging I

- Zuverlässige Nachrichtenübermittlung
- Sender der Nachricht erhält
 - *Acknowledge* als eigene Nachricht
 - oder *Acknowledge* als teil der Antwort vom Empfänger



- Nachrichtensequenz
 - Eingebettet im SOAP Header
 - Eindeutige Identifizierung einer Nachricht
 - Acknowledge pro Nachricht oder für eine Sequenz



- Auslieferungsversicherung
 - AtMostOnce
 - AtLeastOnce
 - ExactlyOnce
 - InOrder



→Reliable Messaging IV

```
<wsrm:Sequence>  
  <wsu:Identifier>http://www.examples.ws/</wsu:Identifier>  
  <wsrm:MessageNumber>6</wsrm:MessageNumber>  
  <wsrm:LastMessage />  
</wsrm:Sequence>
```

```
<wsrm:SequenceAcknowledgement>  
  <wsu:Identifier>http://www.examples.ws/</wsu:Identifier>  
  <wsrm:AcknowledgementRange Upper="3" Lower="1" />  
  <wsrm:AcknowledgementRange Upper="6" Lower="5" />  
</wsrm:SequenceAcknowledgement>
```



→Policy I

- Erstellen und Anwenden von Regeln (Policies)
- Policy Assertions
 - Definition
- Policy Subjects
 - Objekt, auf das Regel angewandt werden soll
- Policy Attachments
 - Verbindet Policy Assertion mit policy Subject



→Policy II

```
<wsrm:Sequence>
  <wsu:Identifier>
    http://www.examples.ws/
  </wsu:Identifier>
  <wsrm:MessageNumber>
    6
  </wsrm:MessageNumber>
  <wsrm:LastMessage />
</wsrm:Sequence>
```

```
<wsp:PolicyAttachment>
  <wsp:AppliesTo>
    <wsrm:SequenceRef>
      <wsu:identifier>
        http://www.examples.ws/
      </wsu:identifier>
    </wsrm:SequenceRef>
  </wsp:AppliesTo>
  <wsp:policy>
    <wsrm:DeliveryMessageAssurance
      Value="wasrm:AtLeastOnce"
      wsp:Usage="wsp:Required" />
  </wsp:policy>
</wsp:PolicyAttachment>
```



→ Attachments I

- Möglichkeit Dateien an SOAP Nachrichten anzuhängen
- 2 konkurrierende Standards
 - WS-Attachment
 - SOAP Message with Attachments (SwA)



- WS-Attachment
 - Primary message part
 - Eigentliche SOAP Nachricht
 - Secondary message part
 - Das Angehängte
 - Kodiert als DIME Type
 - Referenzierung von primary nach secondary und zwischen secondaries möglich



- SOAP Message with Attachments
 - Primary message part
 - Eigentliche SOAP Nachricht
 - Secondary message part
 - Das Angehängte
 - Kodiert als MIME Type